

CLAIMS

1. Process for programming an automation application program on an automation equipment programming station that comprises:

- 5       - a step to define a plurality of structured type objects (1.10, 1.20) each representing an input-output channel of an input-output module of the automation equipment, the definition of each structured type including at least one characteristic element (1.11) corresponding to input-output information  
10       exchanged between the input-output module and the application program, and the relative topological address of the input-output information for each element,
  - 15       - a step to write an application program (10), declaring symbolic input-output variables (100) of the application program (10) as an instance of a previously defined structured type object,
  - 20       - a step to configure symbolic input-output variables comprising a definition of the physical location of each input-output module of the automation equipment associated with the symbolic input-output variables (100) of the application program,
  - 25       - a step (40) to automatically interpret the application program to execute it on the automation equipment, comprising a step to replace symbolic input-output variables (100) in the application program (10) by the complete topological address (200) of the corresponding input-output information.

2. Programming process according to claim 1, characterized in that each symbolic variable (100) of the program comprises two fields, a first field (101) composed of a character string chosen by the application program designer, so that a structured type object can be made to correspond with the symbolic variable (100), and a second field (102) composed of an identification of an element of the structured type object associated with the symbolic variable (100).

10 3. Programming process according to claim 1 or 2, characterized in that the replacement step comprises:

- a step (34) to search for the relative address defined for each structured type element in a table (1.1, 1.2) of elements of a structured type object stored on the programming station,

- a step (32) to search in a configuration table for the physical location declared for each module that the designer has associated with symbolic input-output variables (100) of the application program,

- a step (33, 35) to construct the exact topological address of each symbolic variable (100) of the application program, using interpretation means on the programming station, starting from the relative address and the physical location found.

4. Programming process according to claim 1 or 2, characterized in that the step to define structured type objects comprises a step to create a table (1.1, 1.2) of structured type object elements comprising a first column containing at least one identification of a characteristic data of the structured type object, a

second column containing the elementary data type (EDT) and a third column containing the relative address of the data, and then memorizing this table in portable memory means, for each structured type object.

5           5. Programming process according to claim 3, characterized in that the table (1.1, 1.2) of structured object type elements comprises a fourth column containing a description of the data, and a fifth column containing read or write rights for each  
10 data.

          6. Programming process according to any one of claims 1 to 5, characterized in that the process comprises a step to configure input-output modules comprising a step to select a commercial reference of  
15 an input-output module, and assignment of the selected input-output module to a determined physical location, the interpretation step then including a step to check that the input-output module selected at a determined physical location is compatible with the structured  
20 type object configured at the same physical location.

          7. Programming station for programming automation equipment comprising means of memorization and display, and means of interaction with a designer of an automation application program (10), characterized in  
25 that the programming station comprises an editor of symbolic variables (100) to generate a configuration table (6) stored on the memory means, the programming station also includes several tables (1.1, 1.2) of structured type object elements stored on the memory  
30 means, and means of interpreting an application program

(10) comprising at least one symbolic variable (100) defined by the designer using the editor.

8. Programming station according to claim 7, characterized in that it comprises means of compiling  
5 the application program interpreted by interpretation means to transform the interpreted application program into an automation application that can be executed on an automation equipment

9. Programming station according to claim 8,  
10 characterized in that it comprises means of transferring the executable automation application onto either portable memory means compatible with the automation equipment, or directly onto the memory means of the automation equipment.